UW/BBN Rich Transcription for Conversational Telephone Speech

Mari Ostendorf, Jay Kim, Sarah Schwarm, Bill McNeill

University of Washington

EARS MDE Evaluation Meeting

13-14 Nov 2003

Outline

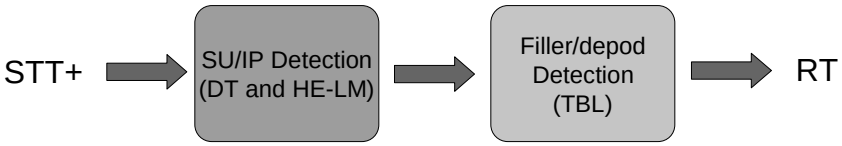
- **Approach**
  - **2-stage detection**
  - **Serial vs. system combination architectures**
- **Summary of eval results (CTS only)**
- **Recent improvements**
- **Error analysis**

RT System Overview

- **BBN provides STT + time alignments + speaker labels**
- **UW feature processing (prosody + lexical)**
- **UW 2-stage detection of structural MDE**
  - **First: find boundary (between word) events: SUs & IPs**
  - **Second: detect depod and filler words**
- **Optional (for integrated system), combine UW & BBN SU prediction, before depod & filler detection**
  - **SU combination work by Amit Srivastava at BBN**

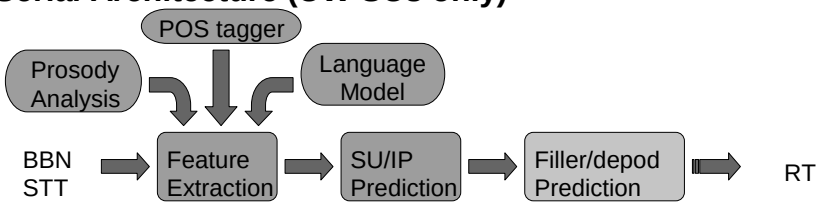
Two-Stage Detection of Structural MDE

- **Detect SUs and IPs together with a decision tree (DT) and a hidden event language model (HE-LM)**
  - **Joint detection because SUs & IPs have some similar acoustic cues but different language cues**
  - **Possible problems when SU & IP co-occur**
- **Detect fillers and edits using the transformation-based learning (TBL) algorithm**
  - **Presence of an IP is useful information, esp. for edit detection**
  - **If IP is known, then acoustic cues are much less important than language cues**

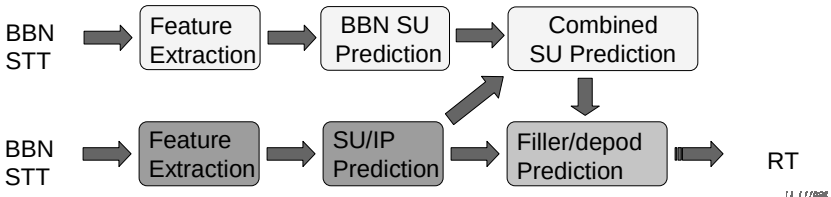


System Architecture

- **Serial Architecture (UW SUs only)**



- **Combined Architecture (UW+BBN SUs)**



Experimental Paradigm

- **Training data**
  - **417 conversations annotated by LDC using V5 spec (LDC1.3)**
  - **1086 conversations from disfluency-annotated Switchboard Treebank data (using Meteer-mapping from SRI-ICSI)**
  - **Reference transcription only**
  - **Investigation of which combination of data is most useful**
- **Development data**
  - **18 Fisher and 18 Swbd conversations, annotated with V5 spec**
- **Scoring tools: both rt-eval & su/df-eval used, but final decisions made based on rt-eval**

Stage 1: Detecting SUs and IPs

- Recognize 4 classes of events: SU, SU-inc, IP, other
- Use decision tree to integrate continuous prosodic cues and symbolic lexical
- Hidden event language model (using SRI LM toolkit)
  - Train trigram LM with tokens representing SU/IP inserted in the word stream
  - During testing, use the LM as a hidden Markov model
    - Consider word/event as states and words as observations
    - Use a forward-backward dynamic programming algorithm to calculate posterior probabilities P(event|words)
- Use hidden event posterior…
  - In decision tree as an additional feature, OR
  - As a separate score in (linear) combination with the decision tree posterior

Decision Tree Features

- Acoustic-prosodic features
  - Average normalized duration over the word and the rhyme of the word, silence duration
  - F0 statistics (min, max, avg, slope) over a word, normalized by speaker statistics and statistics of F0 differences between the current and the following word (use SRI F0 processing & stylization)
  - Energy statistics (min, max, avg) over a word and rhyme, normalized by speaker statistics
  - Word position in the speaker turn and indicators of speaker overlap and start and end of a turn

Decision Tree Features

- **Lexical Features**
  - **Flag indicating whether following words can be fillers**
  - **Posterior probabilities from HE-LM**
  - **Part-of-speech tags, grouped into 15 categories to reduce the cost of training**
  - **Indicator of word and grouped POS tag pattern match across a word boundary, skipping potential filler words**

Word	But	there	are	*you*	*know*	it's	like …
POS	CC	EX	VBP	*PRP*	*VBP*	PRP+BES	IN
Grouped POS	M2	**M1**	**VL**	*M1*	*M1*	**M1+VL**	M2

POS match for the boundary after "are"

Decision Tree Prediction Results

- **On Dev set, reference transcription**
- **Using word-based decision tree metric**
  - **Overall accuracy: 91.1**
  - **Chance: 75.5**

Type	Recall (%)	Precision (%)
SU	**81.5**	**80.4**
ISU	**36.5**	**69.7**
IP	**66.1**	**78.7**

- **Inc-SU is most difficult category**

Stage 2: Identifying Edits and Fillers

- After SUs and IPs are marked, use rules to identify edits and fillers
- Automatic rule design using Transformation-Based Learning (TBL)
  - Brill [Computational Linguistics, 1995]
  - Key features of this rule-learning algorithm:
    - Corpus-based, error-driven automatic learning
    - Simple, concise, comprehensible rules
  - Useful in many NLP problems, e.g. part-of-speech tagging (similar to edit/filler labeling), parsing, spelling correction
  - We used the fnTBL toolkit (Ngai and Florian [ACL, 2001])
    - Advantage: fast training
    - Disadvantage: symbolic but not numeric features

How TBL Works

- Apply an initial tag to each item in the corpus (baseline predictor)
- Repeat:
  - Use templates to generate all possible transformation rules that correct at least one error
  - Score rules using an objective function
  - Choose the best transformation rule and apply it to the corpus
  - Stop when the score of the best rule falls below a threshold
- Need templates that specify allowable rules.
  - Rules consist of a triggering environment and a transformation.
  - Example: (part-of-speech tagging)
    - Template: word_0 word_1 => pos
    - Rule: word_0=table word_1=the => pos=verb

Current TBL Configuration

- Predicted SUs and IPs are added to the data as special "words"
- Baseline predictor: no disfluency (most common case)
- Rule templates consider:
  - Features of the current word and/or neighbors
  - Proximity of potential FP/DM/EET terms
  - Word/POS matches between current and nearby words, e.g.
    - that IP that (word match)
    - the dog IP the cat (POS match)
- Objective function: min token error rate

Features Used in TBL Stage

- Identity of the word (includes SU/IP)
- POS and GPOS (POS group) of the word (same as decision tree features)
- Flags indicating whether the word is commonly used as: filled pause (FP), back channel, explicit edit term (EET), and/or discourse marker (DM)
- Flags indicating whether word/POS/GPOS matches the word/POS/GPOS that is 1/2/3 positions to its right
- Turn and segment boundary flags (same as decision tree features)
- Tag to be learned (FP, EET, DM, depod, and none)

Design Questions and Findings

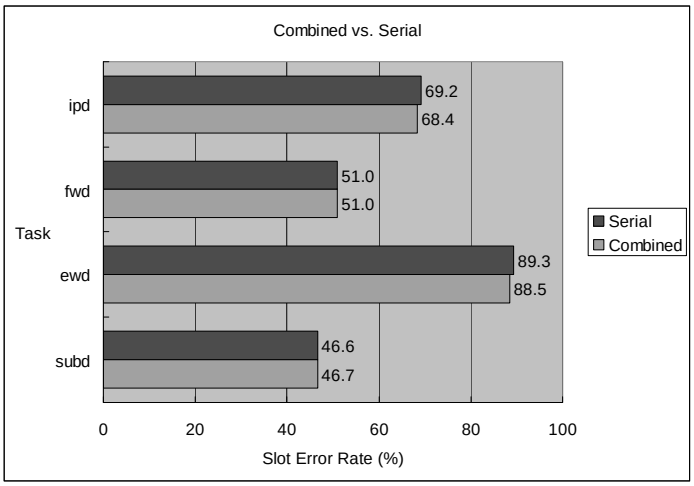
- **Which training data to use for stage 1 vs. stage 2, i.e. is the Meteer-mapped data useful?**
  - **NO for SU/IP detection (problem for IPs is edit recall)**
  - **YES for edits and fillers (reduces insertions)**
- **Should we model all IPs or just IPs associated with edits? (adding filler IPs in post-processing)**
  - **Using all IPs gave better results in SU/IP detection**
  - **Edit detection is slightly better using only edit IPs**

Pre-Evals

- **Should TBL train with hand-labeled or automatically predicted SUs and IPs?**
  - **Small gain from automatically predicted SU/IP**
- **Should HE-LM be used in the decision tree or as a separate knowledge source?**
  - **Mixed results….**

Post-Evals

Eval Result: Combined vs. Serial SUs



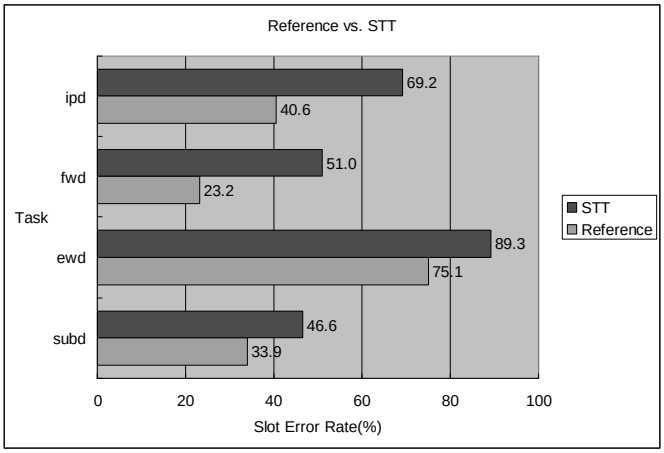
Combined SUs lead to small gain in IP and edit detection

Eval Result: Details for Serial Case

Task	%Corr	%Del	%Ins	%SER
Filler	64.04	35.96	15.07	51.02
DEPOD	26.90	73.10	16.20	89.29
IP	49.55	50.45	18.76	69.20
SU	73.37	23.63	19.97	46.60

- Good news: (relatively) good SU performance
- Bad news:
  - Missing a lot of edits and IPs (related)
  - Filler accuracy is much worse than other sites (why??)

Eval Result: Ref vs. STT Hyp

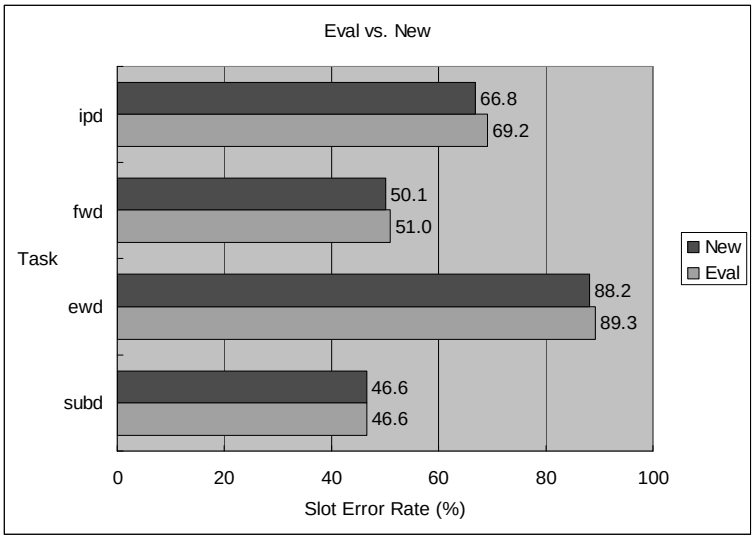


- Note: System processing references is identical to ASR hyp system, so fragments are not used in IP detection.
- Observe biggest loss for fillers and IPs

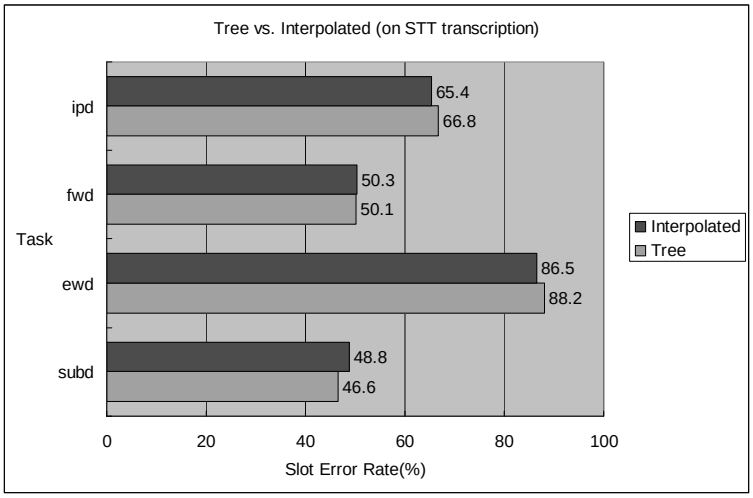
Improvements since Evals

- **Bug fixes:**
  - **Retrained POS tagger with unicased words and with all punctuations stripped out**
  - **Fixed a bug in TBL feature processing**
- **Real improvements:**
  - **Use iterative feature selection to find a more robust set of acoustic-prosodic features**
  - **Trained TBL with predicted SU/IP tags**
- **Combined HLM and DT models by interpolating scores**
  - **Weighting factor was determined empirically to maximize overall accuracy of SU/IP prediction on Dev STT transcription**

Small Gains in Performance



HE-LM in Tree or Interpolated with Tree?



Separating the HE-LM out helps IPs (and hence edits), but hurts SUs.

Known Problems and Error Analysis

- **We never predicted words labeled as both filler and edit (due to a bug in fnTBL)**
  - **These words were treated as fillers in TBL training**
  - **May not be a big problem since only 0.5% of all edit and filler words in LDC1.3 are both filler and edit words.**
- **We never predict boundaries as having both SU and IP**
  - **Treat them as having just SU when training DT, HE-LM, and TBL; insert IPs after fillers are detected in the TBL stage**
  - **In LDC1.3, 12.8% of boundaries that contain SU also have IP**
  - **Insignificant for IPs following edits but 38.6% of IPs before fillers are affected**
- **Problems with fillers:**
  - **Most of our filler errors were due to STT errors**
  - **Most non-speech recognizer filler errors involved the words "So" (at SU start) and "like", which are hard problems**

Known Problems and Error Analysis (cont.)

- **Fragments:**
  - **In LDC1.3, 17.2% of edit IPs have word fragments occuring before them, and 9.9% of depods had just a single fragment.**
  - **In Dev set, 35.5% of edit IPs are associated with word fragments**
  - **IP detection performance was significantly worse for those IPs associated with fragments.**

Percentage of missed IPs on the Dev set:

Transcription	IPs after fragments	Other edit IPs
Reference	81.7	37.6
STT	74.0	51.2

STT can "help" when fragments aren't explicitly modeled, since the fragment is often deleted or recognized as the full word.

Summary

- **2-stage approach:**
  - **Joint SU/IP detection in decision tree, integrates prosody & lexical cues**
  - **TBL predictor for fillers and depods**
- **Some experiment findings**
  - **Despite problems, the Meteer-mapped data has some value**
  - **Acoustic reasons for fillers to have IPs (better SU/IP detection), but using filler IPs in TBL stage hurts filler prediction**
  - **Some gains from SU system combination**
  - **Mixed results on how to integrate LM and prosodic cues**
- **Future work**
  - **Fix known problems**
  - **Further explore system combination**
  - **Integrate parsing into system**

Extra Slides

Key Differences Relative to Others

- **Compared to SRI's work**
  - **We modeled and predicted SUs and IPs together; SRI modeled them separately**
  - **SRI downsampled training data to deal with the imbalanced data and applied bagging techniques in decision tree training**
  - **SRI combined word-based, POS-based and class-based LMs for SU detection**
  - **We used LM scores as features in decision tree training; SRI interpolated the scores.**

Key Differences Relative to Others (cont.)

- **Compared to UMD**
  - **UMD used TBL with features similar to ours to detect DEPOD and fillers. However, they also included prosody-based features in TBL:**
    - **Flag indicating whether a pause follows the current word**
    - **Flag indicating whether the word was used more often than average by the speaker**
- **Compared to CU**
  - **CU used word and class based trigram LMs and a decision tree trained with prosodic features to detect SUs.**
  - **They combined LM and decision tree scores with lattice tools**

Electrical Engineering
University of Washington